Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Please amend the claims as follows:

1.-10. (Cancelled).

11. (Currently Amended) A method comprising:

dividing a dynamic sequential program into multiple epochs comprising a first epoch instance and a second epoch instance, wherein each epoch includes two or more instructions;

in a redundant multi-threading (RMT) system having leading and trailing threads, redundantly executing in parallel the first epoch instance and second epoch instance as the leading and trailing threads, respectively;

for the executed first epoch instance and second epoch instance, saving store results of the first epoch instance and the second epoch instance as speculative stores to memory, the speculative stores being exposed;

comparing the saved exposed stores; and

[[if]]when the exposed stores match, committing a single set of the exposed stores to an architectural memory state corresponding to the dynamic sequential program.

12. (Previously Presented) The method of claim 11, wherein the speculative stores are from a re-order buffer.

13. (Previously Presented) The method of claim 12, wherein the two or more instructions executed in response to the execution of the first or second epoch instances are buffered prior to epoch execution completion.

14. (Cancelled).

15. (Previously Presented) The method of claim 11, wherein the memory is L1 cache memory.

16. (Currently Amended) A method, comprising:

~~breaking a program to be executed into multiple epochs each having two or more instructions;~~

redundantly executing [[the]]a program broken into multiple epochs by redundantly executing each epoch separately, and

sending speculative results for each epoch to memory;

checking the speculative results in memory for each epoch against each other; and

[[if]]when the speculative results match, committing a single set of the speculative results to an architectural memory state corresponding to the program.

17. (Previously Presented) The method of claim 16, in which the speculative results are speculative stores.

18. (Previously Presented) The method of claim 16, in which the memory is L 1 cache memory.

19. (Currently Amended) The method of claim [[11]]16, further comprising committing the store results of [[the]]a first epoch instance or second epoch instance to a sequential architectural state of the computation in response to the first epoch instance or second epoch instance becoming an oldest epoch.

20. (Previously Presented) The method of claim 16, further comprising committing the results of at least one of the multiple epochs to a sequential

architectural state of the computation in response to the at least one epoch becoming an oldest epoch.